Adjustable Autonomy for Human-Centered Autonomous Systems

Gregory A. Dorais [gadorais@ptolemy.arc.nasa.gov][•]
R. Peter Bonasso [bonasso@mickey.jsc.nasa.gov][*]
David Kortenkamp [kortenkamp@jsc.nasa.gov][*]
Barney Pell [pell@ptolemy.arc.nasa.gov][+]
Debra Schreckenghost [schreck@mickey.jsc.nasa.gov][*]

Abstract

We expect a variety of autonomous systems, from rovers to life-support systems, to play a critical role in the success of future space missions. The crew and ground support personnel will want to control and be informed by these systems at varying levels of detail depending on the situation. Moreover, these systems will need to operate safely in the presence of people and cooperate with them effectively. We call such autonomous systems human-centered in contrast with traditional "black-box" autonomous systems. Our goal is to design a framework for human-centered autonomous systems that enables users to interact with these systems at whatever level of control is most appropriate whenever they so choose, but minimize the necessity for such interaction. This paper discusses on-going research at the NASA Ames Research Center and the Johnson Space Center in developing human-centered autonomous systems that can be used for space missions.

Introduction and Motivation

Autonomous system operation at a remote planetary sites provides the crew with more independence from ground operations support. Such autonomy is essential to reduce operations costs and to accommodate the ground communication delays and blackouts at such a site. Additionally, autonomous systems, e.g., automated control of life support systems and robots, can reduce crew workload [Schreckenghost *et al.*, 1998] at the remote site. For long duration missions, however, the crew must be able to (partially or fully) disable the autonomous control of a system for routine maintenance (such as calibration or battery recharging) and occasional repair. The lack of in-line sensors that are sufficiently sensitive and reliable requires manual sampling and adjustment of control for some life support systems. Joint man-machine performance of tasks (referred to as traded control) can improve overall task performance by leveraging both human propensities and autonomous control software capabilities through appropriate task allocation [Kortenkamp, *et al.*, 1997]. The crew may also desire to intervene opportunistically in the operation of both life support systems and robots to respond to novel situations, to configure for degraded mode operations, and to accommodate crew preferences. Such interaction with autonomous systems requires the ability to adjust the level of autonomy during system execution between manual operation and autonomous operation. The design of control systems for such adjustable autonomy is an important enabling technology for advanced space missions.

[•] Caelum Research, NASA Ames Research Center, Mail Stop 269-2, Moffett Field, CA 94035
[*] TRACLabs, NASA Johnson Space Center, TX
[+] RIACS, NASA Ames Research Center, Mail Stop 269-2, Moffett Field, CA 94035

Definition of adjustable autonomy and human-centered autonomous systems

A human-centered autonomous system is an autonomous system designed to interact with people intelligently. That is, the system recognizes people as intelligent agents it can (or must) inform and be informed by. These people may be in the environment of the autonomous system or remotely communicating with it. In contrast, a traditional "black box" autonomous system executes prewritten commands and generally treats people in its environment as objects if it recognizes them at all.

Typically, the level of autonomy of a system is determined when the system is designed. In many systems, the user can choose to run the system either autonomously or manually, but must choose one or the other. Adjustable autonomy describes the property of an autonomous system to change its level of autonomy to one of many levels while the system operates. A human user, another system, or the autonomous system itself may "adjust" the autonomous system's level of autonomy. A system's level of autonomy can refer to:

- how complex the commands it executes are
- how many of its sub-systems are being autonomously controlled
- under what circumstances will the system override manual control
- the duration of autonomous operation

For example, consider an unmanned rover. The command, "find evidence of stratification in a rock" requires a higher level autonomy than, "go straight 10 meters." The rover is operating at a higher level of autonomy when it is controlling its motion as well as its science equipment than when it is just controlling one or the other. The rover overriding a user command that would cause harm or to automatically recover from a mechanical fault also requires a higher level of autonomy than when these safeguards are not in place. Finally, a higher level of autonomy is required for the rover to function autonomously for a year than for a day. Adjustable autonomy is discussed further in [Bonasso *et al.*, 97a][Pell *et al.*, 98].

Manned trip to Mars statistics

The fast-transit mission profile described in [Hoffman and Kaplan, 97] of a hypothetical manned Mars mission sets a launch date of 2/1/2014 arriving on Mars 150 days later on 7/1/2014. After a 619-day stay on Mars, the crew will depart Mars on 3/11/2016 and arrive on Earth after 110 days on 6/26/2016.

Crew time on a Mars mission will be a scarce resource. Figure 1 outlines a possible timeline of how a crew of 8 would allocate their time based on 600-day mission on the Mars surface. The figure is from the NASA Mars Reference mission [Hoffman and Kaplan, 97, p. 1-15] cited from [Cohen, 93]. In this figure, how the 24 (Earth) hours per day per person of the crew are expected to be used are averaged by task classes represented by each column. The production tasks, which differ throughout the mission, are separated into mission phases, represented by rows, of lengths indicated by the Mission Duration column whose total is 600 days.

This schedule is based on a crew of 8. However, we expect the crew size of the first manned Mars mission to be at least 4 but no more than 6. A smaller crew would probably result in a reduction in the total duration of manned surface excursions and an increase in the time devoted to system monitoring, inspection, calibration, maintenance, and repair per person. For example, during a recent 90-day, closed-system test at NASA Johnson Space Center, four crewmembers spent roughly 1.5 hours each day doing maintenance and repair. During this period, they had a continual (i.e., 24-hours per day) ground control presence and engineering support to help in these tasks [Lewis *et al.*, 1998]. This 90-day test used only a small subset of the systems that would be involved in a Mars base. Also, the tasks described in Figure 1 do not include growing and processing food at the Mars base, but assume all food for the entire 600 days is pre-packaged and shipped from earth.

Personal: 14 hr/day					Overhead: 3 hr/day			Production: 7 hr/day	Mission Duration
Sleep, Sleep Prep, Dress, Undress: 8 hr	Hygiene, Cleaning, Personal Communication: 1hr	Recreation, Exercise, Relaxation: 1 hr	Eating, Meal Prep, Clean-up: 1 hr	One Day Off Per Week: 3 hr (per day)	General Planning, Reporting, Documentation, Earth Communication: 1 hr	Group Socialization, Meetings, Life Sciences Subject, Health Monitoring, Health Care: 1 hr	System Monitoring, Inspection, Calibration, Maintenance, Repair: 1hr	Site Prep, Construction	90 days
								Verification	
								Week Off	7 days
								Local Excursions	50 days
								Analysis	
								Distant Excursion	10 days
								Analysis	40 days
								Week Off	7 days
								Distant Excursion	10 days
								Analysis	40 days
								Distant Excursion	10 days
								Analysis	40 days
								Week Off	7 days
								Distant Excursion	10 days
								Analysis	40 days
								Distant Excursion	10 days
								Analysis	40 days
								Flare Retreat	15 days
								Week Off	7 days
								Distant Excursion	10 days
								Analysis	40 days
								Distant Excursion	10 days
								Analysis	40 days
								Week Off	7 days
								Sys Shutdown	60 days

Figure 1. Possible time line for first Mars surface mission

Although 600 days may sound like a long time, the time allocation schedule only has 7 distant excursions. In this schedule, a distant excursion is a multi-day trip on a manned rover to locations within a 5-day drive radius of the Mars base. Of course, the above schedule could be radically changed if a major malfunction were to occur. However, major changes may be necessary simply due to poorly estimating the time needed for the tasks.

Currently for a space shuttle mission, dozens of people are dedicated to system monitoring 24 hours per day. In order for the Mars crew members to achieve the low level of time for system monitoring, inspection, calibration, maintenance, and repair shown in the figure, the systems they use will need to reliably operate with minimal human assistance for nearly two years.

Motivation for autonomy and robots

Given the statistics described in the previous subsection, it is apparent that a successful mission that accomplishes scientific objectives (as opposed to a mission that just keeps the crew alive) will require advanced autonomy, both in terms of controlling the environment and in robots to replace manual labor. Automation and robots provide the following advantages:

- Increased safety of crew
- Increased safety of equipment
- Increased science capabilities
- Increased reliability of mission
- Decreased crew time for monitoring and maintenance
- Decreased mission operation costs
- Decreased demand on deep space ground antennas (DSN)

To illustrate the need for autonomy, consider the following example. Most of us trust the heating of our home to a simple autonomous system. It turns the heating unit on when the temperature sensed by a thermostat falls below an “on” temperature setting and turns the heating unit off when the temperature rises above an “off” temperature setting. We certainly wouldn’t want to burden a crewmember with the need to continually monitor the temperature of each room in the Mars habitat turning on and off the heating unit as necessary if it can be avoided. Temperature control is just one of many tasks that must done continually to keep the crew alive and to enable them to accomplish their science objectives.

Making matters more complex, on Mars many systems will be interdependent making control more difficult than that described in the previous example. In most of our homes, the temperature control system is independent. We can turn the heating unit on and off as we please. One reason this is possible is because we are not constrained by our energy resources. For example, the water heater can be on at the same time as the furnace. We can use a large quantity of energy during the day and not have to worry about not having sufficient energy at night. On Mars, energy will be a major constraint. Using energy for one purpose may mean that energy will not be available for another important task at the same time or later. Crewmembers will not want to have to calculate all the ramifications of turning on a machine each time they do so to avoid undesirable effects. Sophisticated autonomous systems can continually manage constrained resources and plan the operation of various machines so there is not a conflict.

In summary, crew time is scarce on Mars and the system must be closed to calories (i.e., the crew cannot spend more calories producing food than the food provides for them!). The more computers and machines can free up crew time, the greater chance that the mission will meet all of its objectives.

Motivation for adjustable autonomy

It is impossible to predict what will be encountered on a Mars mission. Given the current state-of-the-art in autonomous systems, it is critical that human intervention be supported. Allowing for a an efficient mechanism for crew to give input to an autonomous control system will provide the following benefits to the mission:

- Increased autonomous system capabilities
- Decreased autonomous system development and testing cost
- Increased reliability of mission
- Increased user understanding, control, and trust of autonomous systems

The goal of adjustable autonomy is to maximize the capability but minimize the necessity of human interaction with the controlled systems. Adjustable autonomy makes an autonomous system more versatile and easier for people to understand and to change the system’s behavior.

Consider the following example of a Mars rover. On Mars, the round trip time for a radio signal to Earth can be up to 40 minutes. This is too long to tele-operate a rover from Earth for lengthy missions. During a manned Mars mission, a crewmember can tele-operate a rover. However, when the rover is travelling long distances over relatively uninteresting and benign terrain, a crewmember should not be needed to continually tele-operate the rover. Hence the need for autonomy. Nevertheless, complete autonomy does not solve the problem either. When the rover is traversing the terrain, we expect it will come to places that would be interesting to observe or treacherous to traverse. Unfortunately, commands like, “go out and find interesting images and soil samples for 10 days, then return safely,” are beyond the scope of what autonomous rovers can do for the near future. Even if a rover could reliably execute such commands, sometimes people will want to more directly control the rover to make observations and perform tasks. When a crewmember does take more direct control of the rover, it is desirable to have the rover's autonomous systems make sure the operator doesn't do something by accident, such as drain the batteries so communication is lost or overturn the rover. In some cases, the crewmember may not wish to control the entire rover. Instead, the person will want the rover to continue its current mission and share control of only certain subsystems, such as a camera or scientific instrument. Finally, a high-performance completely autonomous system for a rover is difficult to design and might require too much power to operate the computer required on the rover. For such a system, a significant portion of the development time and the computer resources is required to handle the myriad of situations that a rover is unlikely to encounter. It is simpler to design an autonomous control system that can call on a human for help than one that is expected to handle any situation on its own. Adjustable autonomy supports these capabilities.

Overview of paper

In the remainder of this paper, we describe how adjustable autonomy might affect the crew during a typical day at the Mars base. We then describe autonomous system requirements necessary to achieve the capabilities required. Following the system requirements, we present two systems developed by NASA to perform autonomous system control with adjustable autonomy: Remote Agent and 3T. We describe projects that have or are using these autonomous control systems as well as future NASA projects that may use enhanced versions of these systems. We then conclude.

A day in the life of a mars base

To illustrate the range of operations required of adjustable autonomy systems, we describe a typical day in the life of the crew at a remote Martian site. The remote facility consists of enclosed chambers for growing crops, housing life support systems, conducting experiments, and providing quarters for the crew. There are six crewmembers, each with different specialized skills. These skills include operational knowledge of vehicle hardware, life support hardware, robotic hardware, and control software, as well as mission-specific knowledge required to perform scientific and medical experiments. Similarly, there are a variety of robots with different skills, such as the ability to maneuver inside or outside the remote facility and the ability to mount different manipulators or payload carriers. A joint activity plan specifies crew duties, robotic tasks, and control strategies for life support systems. This plan is based on mission objectives determined by Earth-based ground operations.

At the beginning of each day, the crew does a routine review of the performance overnight of autonomous life support systems. Next the crew discusses the previous day's activities and identifies any activities not completed. Additionally, the crew reviews the ground uplink for changes in mission objectives. Then the pre-planned activities for that day are reviewed and adjusted to accommodate these changes and crew preferences. For example, inspection of the water system may indicate the need to change a wick in the air evaporation water-processing module before the maintenance scheduled later in the week. Alternatively, the results of soil analysis performed by an autonomous rover yesterday may be sufficiently interesting to motivate taking additional samples today. This planning task is performed jointly by the crew and semi-autonomous planning software, where the crew specifies changes in mission objectives and crew preferences and the planning software constructs a modified plan.

In this example, the daily plan includes the following major tasks:

- harvest and replant wheat in the plant growth chamber
- sample and analyze soil at a specified site some distance from the Martian facility
- analyze crew medical information and samples
- repair a faulty circuit at one of the nuclear power plants

Harvesting and replanting are traded control tasks, where the crew lifts trays of wheat onto a robotic transport mechanism that autonomously moves the trays to the harvesting area.

Similarly, trays of wheat seedlings are moved by the robot into the plant chamber and placed in the growth bays by the crew. Sampling soil is an autonomous task performed by the tele-robotic rovers (TROVs). Crewmembers supervise this activity and only intervene when a sample merits closer inspection. This tele-operated sample inspection includes using cameras mounted on the TROV to view samples and manipulators on the TROV to take additional samples manually. Analyzing medical information and samples is a manual task assisted by knowledge-based analysis and visualization software. Repair of the power plants is an EVA task, requiring two crewmembers at the power plant and one crewmember monitoring from inside the surface laboratory. The autonomous power control system detected the fault the previous day and rerouted power distribution around the fault using model-based diagnostic software. The scheduled time required to perform this repair is minimal, because the autonomous diagnostic software has isolated the faulty portion of the circuit before the EVA. Humans are assisted by the tele-operated rovers, which carry tools and replacement parts to the plant and perform autonomous setup and configuration tasks in preparation for the repair.

This example of a daily plan also includes the following routine activities:

- sensor inspection of the plants
- change out of the wick in the air evaporation water processing module
- Human health activities such as exercise, eating, and sleeping.

Sensor inspection of the plant growth chamber is an autonomous robot task. A mobile robot with camera and sensor wand mounted on it traverses the plant chamber and records both visual data and sensor readings for each tray of plants in the chamber. This information is transmitted to the crew workstation and anomalous data are annunciated to the crew for manual inspection. A report summarizing the results of the inspection is generated automatically and reviewed later in the day by the crew. The change out of the wick is a manual task. The autonomous control software for the water system, however, operates concurrently with this operation and annunciates any omissions or misconfigurations in this manual operation (such as failing to turn off the air evap before removing the wick).

In addition to the tasks listed above, the adjustable autonomy system will also be performing its routine tasks of keeping the base operational. That means, monitoring all aspects of the base (water, gases, temperature, pressure, etc.) and operating valves, pumps, etc..., to maintain a healthy equilibrium. These activities happen without crew intervention or knowledge as long as all of the systems remain nominal. They run 24 hours a day, 7 days a week for the entire mission duration.

This simple scenario illustrates a rich variety of man-machine interactions. Humans perform tasks manually, perform tasks jointly with robots or automated software, supervise automated systems performing tasks, and respond to requests from automated systems for manual support. Automated software fulfills many roles including:

- activity planning and scheduling
- reactive control
- anomaly detection and alarm annunciation
- fault diagnosis and recovery

Effectively supporting these different types of man-machine interaction requires that this automated software be designed for adjustable levels of autonomy. It also requires the design of multi-modal user interfaces for interacting with this software. We describe the requirements for adjustable autonomy software that robots and other systems in the next section.

Requirements for a system like above

A life support system such as that described in the previous section will require intelligent planning, scheduling and control systems that can rigorously address several problems:

- Life support systems have difficult constraints including conservation/transformation of mass, crew availability, space availability and energy limitations.

- Life support systems are non-linear, respond indirectly to control signals and are characterized by long-term dynamics.
- Life support systems must react quickly to environmental changes that pose a danger to the crew.
- Life support systems are labor-bound, meaning that automation must provide for off-loading time intensive tasks from the human crew.
- Life support systems must be maintained and repaired by the crew, requiring that autonomous systems be able to maintain minimal life support during routine maintenance and be able to adjust the level of autonomy for crew intervention for repair or degraded mode operation.

Each of these five areas is mission-critical, in the sense that an autonomous system must deal with all of them to ensure success. Dealing with the first problem requires reasoning about time and other resources, and scheduling those resources to avoid conflicts while managing dynamic changes in resource availability (e.g., decreasing food stores). Dealing with the second problem requires the ability to plan for a set of distant goals and adapt the plan, on-the-fly, to new conditions. Dealing with the third problem requires tight sense-act loops that maintain system integrity in the face of environmental changes. Dealing with the fourth problem requires an integration of robotic control and scheduling with the overall monitoring and control of the life support system.

To support these goals, we have identified a core set of competencies that any autonomous system must have. These are described in the following subsections.

Sensing/Actuation/Reactivity

The autonomous system needs information about the environment so that it can take actions that will keep the system in equilibrium. Environmental sensors connected to control laws, which then connect to actuators, are critical to the success of an autonomous system. The control laws are written to keep a certain set point (for temperature, light, pH, etc.) and run at a high frequency, continually sampling the sensors and taking action. They form the lowest level of autonomous control. Of course, most control laws work only in very specific situations. As the environment approaches the boundaries of the control law's effectiveness, the system can become unstable. A key component of a reactive control law is *cognizant failure*, that is, the reactive control law should know when the environment has moved out of its control regime and alert a supervisory module that can make appropriate, higher-level adjustments [Gat, 1998].

Sequencing

Reactive control alone will not result in a stable system. Certain routine actions will need to be taken at regular intervals. These routine actions involve changing the reactive control system (for example, to switch from nighttime to daytime). This kind of control is the job of a *sequencer*. Of course, changes to the reactive control system may not have their intended effects. For this reason, the sequencer needs to *conditionally* execute sequences of actions. That is, the sequencer constantly checks the sensor values coming from the environment and decides its course of action based on that data. For example, the sequencer may begin heating up a chamber by turning a heater on. If the chamber temperature does not rise, the sequencer could respond by turning a second heater on. This second action is conditional on the result of the first action. A sequencer is also needed to insure that interacting subsystems are coordinated and do not conflict, for example, when they both require a common resource at the same time.

Planning/Scheduling

Intelligent planning software is used to construct dynamically task sequences to achieve mission objectives given a set of available agents (crew, robots) and a description of the facility environment (the initial situation). This can include both periodic activities to maintain consumables (like oxygen, food) and the facility (maintenance), and aperiodic activities (like experiments). Periodic activities would include planting and harvesting crops, processing and storing food, maintaining crew health (through exercise, meals, and free time), and maintaining robots, facility systems (like life support), and transportation systems. Aperiodic activities would include responding to system anomalies, robot and system repair, and scientific experimentation. Some of these activities must be prepared for weeks or even months ahead. Crop planning to maintain balanced food stores must be done well in advance due to lengthy crop growth time (e.g., 60-80 days for wheat). The usage of other consumables (like water, oxygen, carbon dioxide) must be monitored and control strategies altered to adjust for mass imbalances or product quality changes. Because of the complexity of such a plan, it should be generated iteratively, with interim steps for the crew to evaluate the plan "so far" and with support for changing goals and constraints to alter the plan at these interim points. The crew should be able to compare alternative plans resulting from these changes and from different optimization schemes, and to specify one of the alternatives for use or additional modification. They should be able to compare previous plans to current plans to identify similarities and highlight differences. The new plan will include activities affecting the control of robots and life support systems within the facility. Portions of the plan related to such control must be provided to the control software for execution.

Intelligent scheduling software is used to refine the task sequences generated by the planning software. Specifically, the scheduler will be used to adjust the time resolution between early and later versions of a plan. It will be used to generate alternative plans, optimized along different dimensions (e.g., optimized for minimum elapsed time, optimized for maximum usage of robots). A system, such as the one described in the previous section, often has several different dimensions that must be optimized. A scheduling system must allow users to understand the interactions and dependencies along the dimensions and to try different solutions with different optimization criteria.

Abstraction is also critical to the success of planning and scheduling activities. In our scenarios, the crew will often have to deal with planning and scheduling at a very high level (e.g., what crops do I need to plant now so they can be harvested in six months) and planning and scheduling at a detailed level (e.g., what is my next task). The autonomous system must be able to move between various time scales and levels of abstraction, presenting the correct level of information to the user at the correct time.

Model-based diagnosis and recovery

When something goes wrong, a robust autonomous should figure out what went wrong and recover as best as it can. A model-based diagnosis and recovery system, such as Livingstone [Williams and Nayak, 96], does this. It is analogous to the autonomic and immune systems of a living creature. If the autonomous system has a model of the system it controls, it can use this to figure out what is the most likely cause that explains the observed symptoms as well as how can the system recover given this diagnosis so its mission can continue. For example, if the pressure of a tank is low, it could be because the tank has a leak, the pump blew a fuse, a valve is not open to fill the tank or not closed to keep the tank from draining. However, it could be that the tank pressure is not low and the pressure sensor is defective. By analyzing the system from other sensors, it may say the pressure is normal or suggest closing a valve, resetting the pump circuit breaker, or requesting a crewmember to check the tank for a leak.

Simulations

Both the crew and the intelligent planner use simulations of the physical systems. They allow for either the crew or the planner to play *what-if* games to test different strategies (i.e., what if I planted 10 trays of wheat and 5 trays of rice? What would my yield be? How much oxygen would be produced? How much water would it take?). Simulations can also allow for verification of the actual responses of the system to the predicted responses based on the simulation. Any inconsistencies could be flagged as a possible malfunction. For example, if turning on a heater in the simulation causes a rise in temperature, but turning on the same heater in the actual system does not, then the heater or temperature sensor may be faulty. There does not need to be one, complete, high-fidelity simulation of the entire system. Each subsystem may have its own simulation with varying levels of fidelity.

User interfaces

Supervising autonomous software activities must be a low cost human task that can be performed remotely without vigilance monitoring. Providing the human supervisor with the right information to monitor autonomous software operations is central in designing effective user interfaces for supervisory control. The supervisor must be able to maintain with minimum effort an awareness of autonomous system operations and performance, and the conditions under which operations are conducted. She must be able to detect opportunities where human intervention can enhance the value of autonomous operations (such as human inspection of remote soil samples) as well as unusual or unexpected situations where human intervention is needed to maintain nominal operations. Under such conditions, it is important to provide status that can be quickly scanned about ongoing autonomous operations and the effects of these operations on the environment. Notification of important events, information requests, and anomalies should be highly salient to avoid vigilance monitoring. Easy access is needed to situation details (current states, recent activities, configuration changes) that help the supervisor become quickly oriented when opportunities or anomalies occur [Schreckenghost and Thronesbery, 1998].

When interaction is necessary, mixed initiative interaction provides an effective way for humans and autonomous software to interact. A system supports mixed initiative interaction if both the human and the software agent have explicit (and possibly distinct) goals to accomplish specific tasks and the ability to make decisions controlling how these goals will be achieved [Allen, 1994]. There are two types of tasks where mixed initiative interaction with autonomous control software is needed: during joint activity planning and during traded control with robots. For joint plan generation, the human and planning software interact to refine a plan iteratively. The human specifies goals and preferences, and the planner generates a plan to achieve these goals [Kortenkamp, et al., 1997]. The crew evaluates the resulting plan and makes planning trade-offs at constraint violation and resource contention. The planner uses modifications from the crew to generate another plan. This process continues until an acceptable plan is generated. For traded control, the human and robot interact when task responsibility is handed over (when control is “traded”). Coordination of crew and robot activities through a joint activity plan reduces context registration problems by providing a shared view of ongoing activities. To accommodate the variability in complex environments, it is necessary to be able to dynamically change task assignment of agents. Finally, as crew and robot work together to accomplish a task, it is important to maintain a shared understanding of the ongoing situation. The robot must be able to monitor the effect of human activities. This may require enhanced sensing to monitor manual operations and situated memory updates. The crew must be able to track ongoing activities of the robot and to query the robot about its understanding of the state of the environment.

To enable such interaction, the user interface software must integrate with each process in the control architecture to provide the user with the capability to exchange information with and issue commands to the control software. Software provided for building the user interface includes:

- communication software to exchange data and commands with the control architecture
- software for manipulating data prior to display
- software for building a variety of display forms

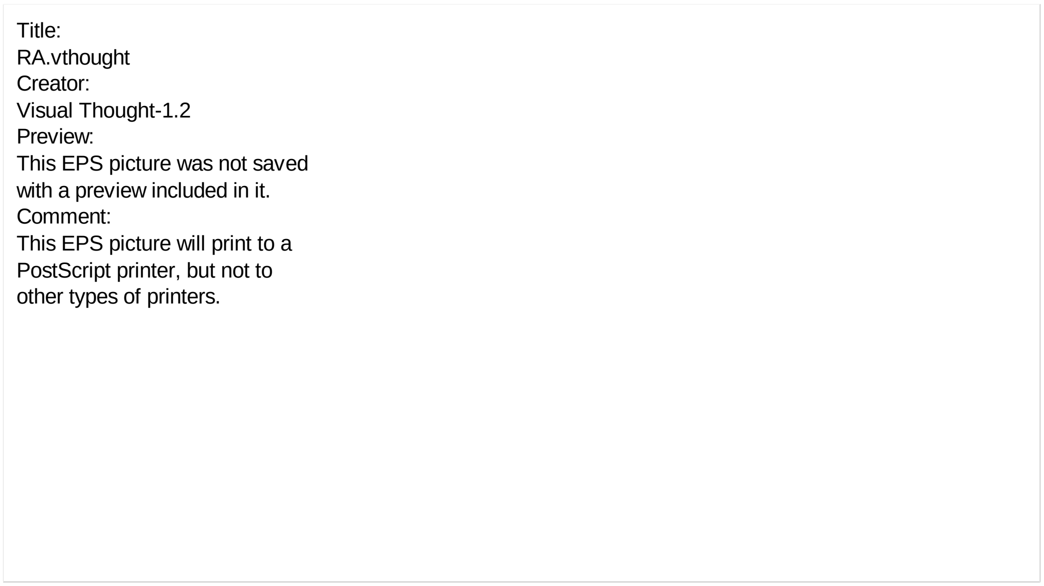
In addition to screen-based, direct manipulation graphical user interfaces, other modalities of interaction will be important in space exploration. Natural language and speech recognition are needed for tasks where the crews’ hands are otherwise occupied or where interaction with a computer is not well-suited to the ongoing task (such as extra-vehicular activity (EVA) tasks or joint manipulation tasks with a robot). Alternative pointing mechanisms such as gestural interfaces also enable multi-modal interaction. Interaction with virtual environments to exercise control in the real environment can improve tele-operations such as remote soil sampling using a TROV or remote robotic maintenance and repair tasks. Visualization software can help manage and interpret data from experiments.

Some current implementations and tests

NASA has developed and implemented two “state-of-the-art” autonomous robotic control systems of the type described in the last section: the Remote Agent and 3T. Each of these architectures is briefly described in this section along with the descriptions of their implementations.

Remote Agent

The Remote Agent (RA) is an agent architecture designed to control extraterrestrial systems autonomously for extended periods. The RA architecture is illustrated in Figure 2.



Title:
RA.vthought
Creator:
Visual Thought-1.2
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Figure 2. Remote Agent architecture

RA consists of four components: Mission Manager (MM), Planner/Scheduler (PS), Model-based Mode Identification and Recovery (MIR), and the Smart Executive (Exec). These components function as follows. Ground control sends the MM a *mission profile* to execute. A mission profile is a list of the goals the autonomous control system is directed to achieve over what may be a multi-year period. When Exec needs a new plan to execute, it requests a plan from MM. MM breaks the mission up into short periods, e.g., two weeks, and submits a plan request to the Planner/Scheduler to create a detailed plan for the next period as requested by Exec. MM insures that resources, such as fuel, needed later in the mission are not available for use in earlier plans. PS creates the requested plan and sends it to Exec to be executed. PS creates flexible, concurrent temporal plans. For example, a plan can instruct that multiple tasks be executed at the same time, but be flexible to when the tasks start and finish. Exec is designed to robustly execute such plans. Exec is a reactive, plan-execution system responsible for coordinating execution-time activities, including resource management, action definition, fault recovery, and configuration management [Pell *et al*., 99].

In the real world, plan execution is complicated by the fact that subsystems fail and sensors are not always accurate. When something does go wrong, is can be difficult to quickly figure out what went wrong and what to do about it. MIR addresses this problem. MIR is a discrete, model-based controller that uses a declarative model of the system being controlled, e.g., a spacecraft. It provides an abstract level of the state of the spacecraft based on its model to Exec. For example, a spacecraft sensor may indicate that a valve is closed when it is, in fact, open. MIR reasons from its model and other sensor readings that the valve must be open. MIR reports to Exec the valve is open and Exec acts accordingly. Another function of MIR is to offer recovery procedures to Exec when Exec is unable to accomplish one of its tasks. Continuing the valve example, let us say that

Exec requires the valve to be closed, but the command to close the valve does not work. Exec sends a request to MIR to figure out how to close the valve without disrupting any other tasks currently being executed. MIR would consult its model of the spacecraft and might instruct Exec to reset a valve solenoid circuit breaker, or to close another valve that will have the effect of closing the desired valve.

RA is discussed further in [Muscettola *et al.*, 98]. We continue by describing systems that have implemented a RA or some of its components.

Saturn Orbit Insertion Simulation

The Saturn Orbit Insertion (SOI) simulation pertains to the phase of a Cassini-like mission where the spacecraft must autonomously decelerate to enter an orbit around Saturn. In addition to handling the nominal procedures of taking images of Saturn's rings as it performs the SOI maneuver, it had to handle faults in real-time in a flight like manner. In the scenario, the main engine overheats when it starts its "burn." The autonomous system shuts down the engine immediately to prevent damage to the spacecraft. However, now the spacecraft is in danger of flying past Saturn. The spacecraft is too far from Earth to wait for a command. The autonomous system starts the backup engine that it previously prepared just in case it was needed. The onboard planner quickly prepares an updated plan. The plan is executed as soon the spacecraft cools sufficiently. During the maneuver, a gyroscope fails to generate data. The system has also prepared for this problem because the backup gyroscope was already warmed up in case it was needed. Additional problems such as coordinating tasks so that there were no power overloads were also handled The SOI is described in more detail in [Pell *et al.*, 96]. This scenario demonstrates that an autonomous can prepare for and react to situations that a human might overlook or react to slowly to even if the person were onboard.

DS1

Deep Space One (DS1) is the first of a series of low-cost unmanned spacecraft missions whose mandate is to validate new spacecraft technologies. NASA scheduled the DS1 launch in October 1998. The science objective of DS1 is to approach then image an asteroid and a comet. An ion propulsion system and the Remote Agent autonomous control system are among the technologies being validated.

Figure 3. Drawing of Deep Space 1 Spacecraft (DS1) encountering a comet

The DS1 Remote Agent will control DS1 for a 12-hour period and a 6-day period during the primary phase of the mission. During these periods, the Remote Agent will generate and execute plans on the spacecraft and recover from simulated spacecraft faults. These faults include a power bus status switch failure, a camera that cannot be turned off, and a thruster stuck closed. The DS1 Remote Agent will periodically generate plans as necessary based on the mission profile (goals & constraints) and the current state of the spacecraft. Model-based failure detection and recovery will be demonstrated. When the model-based recovery system cannot correct the problem, the planner will generate a new plan based on the diagnosed state of the spacecraft. [Bernard *et al.*, 98] discusses the DS1 Remote Agent in depth. The DS1 Remote Agent supports adjustable autonomy by allowing the spacecraft to be controlled as follows:

- entirely from the ground using traditional command sequences
- partially from the ground by uplinking conditional command sequences to be executed with model-based recoveries performed as needed
- autonomously with various ground commands being executed while a plan is also being executed

- completely autonomous with no ground interaction – plans generated onboard as needed

IpexT

IpexT (Integrated Planning and Execution for Telecommunications) is an autonomous control system prototype for managing satellite communications, particularly in crises. The autonomous control system prototype was developed using the RA Planner/Scheduler and Smart Executive. [Plaunt and Rajan, 98] discuss this system in more detail. This system supports adjustable autonomy by operating autonomously in normal circumstances, but permitting operators to take control at various levels, from satellite beam positioning to change the bandwidth of a region to the call priority and quality of a specific call.

Mars Rover Field Test

For this experiment, the Remote Agent Smart Executive and Model-based Diagnosis and Recovery components are being used to control a Mars rover prototype based on the Russian-built rover Marsokhod.

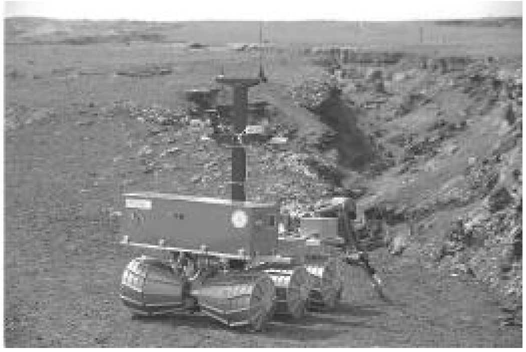


Figure 4. Marsokhod Rover

The field test is scheduled for one week in November 1998 in rough desert terrain. The rover will be commanded by scientists to achieve various science goals by sending high-level commands to the Smart Executive (Exec). Exec will execute conditional plans and manage resources such as power so that the scientists do not have to. When performing sampling and science tests, scientists will share control of the rover with the Remote Agent.

3T

NASA Johnson Space Center and TRACLabs/Metrica Incorporated have, over the last several years [Bonasso *et al* 97b], developed a multi-tiered cognitive architecture that consists of three interacting layers or tiers (and is thus known as 3T, see

Figure **5**).

- A set of hardware-specific control skills that represent the architecture's connection with the world. Control skills directly interact with the hardware to maintain a state in the environment. They take in sensory data and produce actions in a closed control loop. For example, a skill may adjust the flow of base or acid to maintain a pH level. A program called a *skill manager* schedules skills on the CPU, routes skill data and communicates with the sequencing layer of the architecture.
- A sequencing capability that differentially activates control skills using different input parameters to direct changes in the state of the world to accomplish specific tasks. For example, the sequencer may adjust the pH set point of a pH control skill based on overall environmental conditions. We are using the Reactive Action Packages (RAPs) system [Firby, 1989] for this portion of the architecture.
- A deliberative planning capability which reasons in depth about goals, resources and timing constraints. 3T currently uses a state-based non-linear hierarchical planner known as AP [Elsaesser and MacMillan, 1991]. AP determines which sequences are running to accomplish the overall system goals.

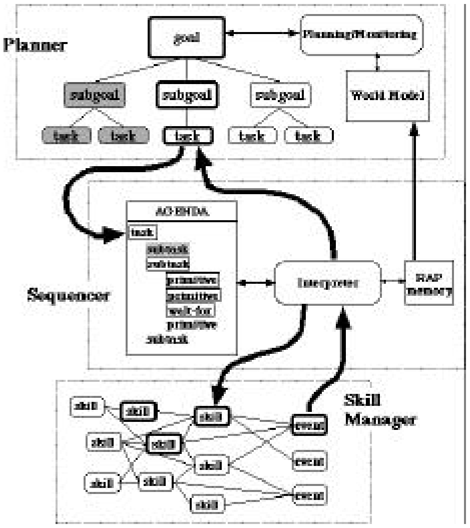


Figure 5. 3T architecture

The architecture works as follows. The deliberative layer begins by taking a high-level goal and synthesizes it into a partially ordered list of operators. Each of these operators corresponds to one or more RAPs in the sequencing layer. The RAP interpreter (sequencing layer) decomposes the selected RAP into other RAPs and finally activates a specific set of control skills in the skills layer. Also activated is a set of event monitors that notifies the sequencing layer of the occurrence of certain world conditions. The activated control skills will move the state of the world in a direction that should cause the desired events. The sequencing layer will terminate the actions, or replace them with new actions when the monitoring events are triggered, when a timeout occurs, or when a new message is received from the deliberative layer indicating a change of plan.

3T Autonomous Control System for Air Revitalization: Phase III test

A series of manned tests that demonstrate advanced life support technology were conducted at the Johnson Space Center (JSC) under the Lunar/Mars Life Support Technology Project. The Phase III test, the fourth in this series of tests, was conducted in the fall of 1997. During the Phase III test, four crewmembers were isolated in an enclosed chamber for 91 days. Both water and air was regenerated for the crew using advanced life support systems. One of the innovative techniques demonstrated during this test is the use of plants for converting carbon dioxide (CO2) produced by crew respiration and solid waste incineration into oxygen (O2). Because the crew and the plants were located in different chambers, it was necessary to build the product gas transfer system to move gases among the crew chamber, the plant chamber, and the incinerator.

Figure 6 shows the physical layout of the product gas transfer system.

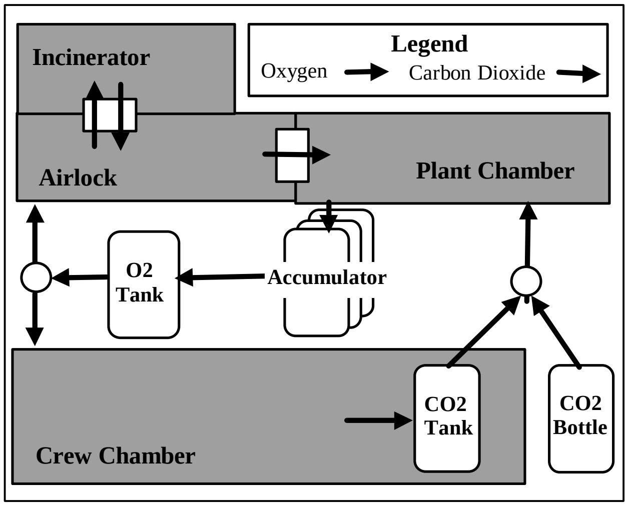


Figure 6. Physical layout of the Product Gas Transfer System

An objective for the Phase III test was to demonstrate that automated control software can reduce the workload of test article engineers (TAEs). We developed an automated control system for the product gas transfer system using the 3T control architecture [Schreckenghost *et al.*, 98a]. The planning tier implements strategies for managing contended resources in the product gas transfer system. These strategies are used (1) to manage the storage and use of oxygen for the crew and for solid waste incineration, and (2) to schedule the airlock for crop germination, planting, and harvesting and for waste incineration. The reactive sequencing tier implements tactics to control the flow of gas. Gas is transferred to maintain O2 & CO2 concentrations in the plant chamber and to maintain O2 concentration in the airlock during incineration. The sequencer also detects caution & warning states and executes anomaly recovery procedures. The skill management tier interfaces the 3T control software to the product gas transfer hardware and archives data for analysis

The 3T product gas transfer control system operated round-the-clock for 73 days. It typically operated with limited intervention by the TAEs and significant engineer workload reduction was demonstrated. During previous tests, TAEs spent at least 16 hours a day at the life support systems control workstation. Product gas transfer required 6-8 hours per week of shift work with 6 hours for each incineration (conducted every 4 days) and 3 hours for each harvest (conducted every 16-20 days).

Even with such minimal human intervention, it was important to design the product gas transfer control system for adjustable autonomy. We designed the system so that a human can replace each tier of automated control. Either the human or the planner can specify the control strategies for the top tier. For the middle tier, either the human or the sequencer can sequence the control actions. This design was useful during phased integration of the control system, allowing each tier to be integrated from the bottom up (starting with the skill manager). It also supported manual experimentation with novel control tactics during the test. A TAE could temporarily disable the automatic control of either oxygen or carbon dioxide transfer, manually reconfigure gas flow, then return to autonomous control. Manual control is executed through the autonomous software and this software does not overwrite human commands while manual control is active. Control setpoints and alarm thresholds are parameterized for manual fine-tuning of control strategies. For long duration tests like the Phase III test, it is important to design control systems to continue operating when control hardware is maintained and repaired. The product gas transfer control system adapts control automatically when sensors are taken out of the control loop manually for calibration or repair. Finally, the autonomous system can initiate a request for information that is only available from a human. For example, the autonomous software requests the TAEs for the results of lab analyses when such information is not available from in-line sensors.

Node 3

Node 3, to be launched in 2002, serves as a connecting module for the U.S. Habitation module, the U.S. crew return vehicle and future station additions. It contains two avionics racks and two life support racks. The crew and thermal systems division at JSC have been developing advanced water and air recovery systems which would be more efficient in terms of power and consumables than those life support systems originally included in node 3. A biological water processing system and a system to recover oxygen from CO2 via a water product are two such advanced systems.

This advanced life support comprises 8 subsystems that must be carefully coordinated to balance the gas and mass flows to be effective. Moreover, the human vigilance required must be minimal. To those ends, JSCs automation and avionics divisions are configuring the 3T control system to run this advanced life support. During ground testing, however, a variety of control techniques are being studied and this requires the ability to vary the autonomy of various subsystems. Such adjustable autonomy is integral to 3T.

Space shuttle Remote Manipulator System Assistant

3T [Bonasso *et al.*, 97] is being used as the software framework for automating the job of NASA flight controllers as they track procedures executed by on-orbit astronauts. The RMS Assistant (RMSA) project focuses on automating the procedures relating to the shuttle's Remote Manipulator System (RMS) and is a pathfinder project for the automation of other shuttle operations. The RMSA system is designed to track the expected steps of the crew as they carry out RMS operations, detecting malfunctions in the RMS system from failures or improper configurations as well as improper or incomplete procedures by the crew. In this regard, it is a “flight controller in a box”. Moving the flight controller functions on-orbit is part of a larger program of downsizing general space shuttle operations. The 3T architecture was designed for intelligent autonomous robots, but has an integral capability that allows adjustable autonomy to include full tele-operation. In that mode, the software acts as a monitoring system, and thus provides an “assistant” framework now and accommodates operations that are more autonomous in the future.

The RMSA was used to “flight follow” portions of the RMS checkout operations in shuttle flights STS80 (November 1996) and STS82 (February 97) as well as various RMS joint movements during payload deployment and retrieval [Bonasso *et al.*, 98]. To flight follow, RMSA was run on the ground with access to the telemetry downlist, but without the crew's knowledge. Additional monitor RAPs were written to “watch” for certain telemetry cues that would indicate that the crew had begun each procedure. The RMSA showed that it could follow crew operations successfully even in the face of loss of data due to communications exclusions or procedures being skipped by the crew.

After the flight following, we held several demonstrations for various RMS-trained astronauts in the spring of 1997. These demonstrations used the RMS simulator to show RMSAs ability to guide checkout procedures, monitor payload deployment and retrieval and to handle off-nominal operations in either tele-operation, semi-autonomous and autonomous modes. The general reaction of the crew was positive, but since it would be sometime before the orbiter's avionics were upgraded to allow autonomous operations, the crew tasked us with developing the tele-operation interface. After a year of development, we began a series of training demonstrations with the crew this past spring that included displays of the RMSA task agenda, expected switch and mode settings, and an integrated VRML 3D synoptic display of the orbiter and RMS positions [3]. This combination was hailed by the crew as the best technology suite for use not only in shuttle RMS operations, but for space station assembly as well. We are currently exercising RMSA for the complete RMS Checkout procedures using mission control simulations in preparation for two Extended Mission Capability flights this winter.

Future projects

The manned Mars missions require systems that can operate autonomously for extended periods as well as operate as in conjunction with people when necessary. Thus, these missions require autonomous software that has the strengths of both the Remote Agent and 3T. We are considering using adjustable autonomy on several projects that we briefly describe below. By enhancing our autonomous control software on the following projects, we are preparing the autonomous control technology needed for manned Mars missions. In doing so, the manned Mars mission will not have to incur the cost or suffer the delay of directly developing and validating these technologies.

Mars TransHab

The TransHab will provide the living and working space for the crew while in transit between the Earth and Mars. The design incorporates a central structural core with an inflatable outer shell. The central core is the structural backbone of the vehicle and provides a mounting surface for all required equipment and systems. The inflatable shell is packaged around the core so that the whole package can be launched in the shuttle. Once in orbit, the shell is inflated, providing most of the interior volume as well as micrometeoroid protection and thermal insulation. The TransHab is one vehicle for the overall architecture required for the Mars mission and has the largest impact on the overall mission mass since it must to go to Mars and back.

Several of the life support systems developed by JSCs crew and thermal system division for node 3 have even more advanced counterparts which will be light-weight, designed to fit in the TransHab core, and run in TransHab's energy-rich environment

In some Mars mission scenarios, a second TransHab will be placed in Mars orbit two years before it is occupied by the crew on the return trip to the Earth. During that time, the life support must be maintained in a stand-by configuration and then respond to the specific crew requirements that may be quite different from what was planned when the mission began.

3T is envisioned for the TransHab control system. Its adjustable autonomy capability will be a primary asset to allow the planned crew of six to adjust the control schemes when encountering unexpected regimes in the 200-day transit to Mars. As well, the adjustable autonomy will allow the crew to set changed, possibly novel activity schedules and profiles for the life support functions on the return trip.

BioPlex

The 3T control architecture has been selected for controlling computer-controlled machines (robotic and regenerative life support) in the BIOPlex facility to be completed at NASA JSC in 2000. The BIOPlex facility will be a ground-based, manned test facility for advanced life support technology destined for use in lunar and planetary bases, and planetary travel (such as Mars TransHab Project). It consists of five connected modules: two plant growth chambers, a crew habitation module, a life support module, and a laboratory. Regenerative life support systems include water recovery, air revitalization, solid waste management, and thermal/atmospheric control. Plant support systems include nutrient delivery, gas management, and thermal/humidity control. Robotic systems include transport, manipulation, and sensor/video scanning. Controlling these heterogeneous systems to maintain food supplies, water, and gas reservoirs, while minimizing solid waste reservoirs (inedible biomass and fecal matter), poses a challenging set of problems for planning and scheduling. The planner must balance conflicting system needs and account for cross system coupling, at time scales varying from hours to months. In this facility, human and robots will jointly execute tasks and must coordinate their efforts. A common/shared schedule for both crew and computer-controlled machines is needed to guarantee such coordination. This schedule must be sufficiently flexible to adapt to crew preferences while stable and robust for computer control. An integrated planning, scheduling, and control architecture that includes both fine time grain scheduling and optimization as well as long term crop planning will be required for BIOPlex. A more complete description of this on-going work is available in [Schreckenghost *et al.*, 1998].

Mars Rover

Although a manned Mars mission may be more than a decade away, NASA has already planned rover missions for launch in 2001, 2003 and 2005. The rover mission in 2001 is likely to be similar to the Sojourner rover, which operated on Mars in 1998 and had very limited autonomous capabilities. However, the goals for the 2003 and 2005 rovers are much more ambitious. Both will travel distances approaching 10km. To achieve this, the rovers will operate completely autonomous while traversing much of Mars. In case of failures, the rovers are expected to autonomously recover. When the rovers reach points selected by scientists, commands from scientists will be uplinked to the rovers to be executed robustly. For example, a high-level command sequence from a scientist might be: if you have the time and the energy to meet your other mission goals, go to this rock, remove the dust, and put the Alpha Proton Xray Spectrometer at this point (as selected by pointing to an image of the rock). If you are collecting valid data, continue collecting data for 2 hours, otherwise continue your mission.

During any day, the rover will communicate with ground for only a couple of hours. To maximize the ability for scientists to perform science, the rover should be at “interesting spots” during the periods and must plan its day accordingly. The scientists would like to be able see the telemetry sent by the rover and respond that same day rather than have to wait for the next day in order to get the commands exactly right to accomplish their science goals and safeguard the rover. By supporting high-level commanding with model-based recoveries, resource management, and autonomous planning, adjustable autonomy addresses this need.

Mars In-situ Propellant Production

A novel approach for reducing the mass of a mission that involves returning from Mars is in situ propellant production (ISPP) on Mars. That is, an ISPP reactor uses CO2 from the Martian atmosphere, hydrogen brought from Earth, and energy (solar or nuclear) to produce the methane and oxygen that will fuel the return vehicle as well as oxygen for life support. In the Mars Reference Mission [Hoffman and Kaplan, 97], an ISPP system runs autonomously for nearly two years before the arrival of the first crewmember on Mars. Its purpose is to produce the fuel for the primary manned-return vehicle. The ISPP system then runs for two more years to produce the fuel for the backup manned-return vehicle. In order to test the reasonableness of this approach, a Mars Sample Return mission that uses an ISPP to fuel the unmanned return vehicle is under study. The proposed launch date is in 2005. In this unmanned mission, an ISPP system, a rover, and a Mars Ascent Vehicle (MAV) will be sent to Mars. While the ISPP system is fueling the MAV, the rover is collecting rock and soil samples that the MAV will return to Earth (or take into Mars orbit to be returned to Earth by another spacecraft).

One of the autonomy challenges ISPP presents is dealing with slowly degrading performance and long-term goals. In addition to failures that may happen suddenly, the performance of the ISPP is expected to degrade over time for several reasons (primarily contamination). Therefore, the rate at which propellant accumulates will decrease over time in a way that may be difficult to predict. The autonomous system must make decisions about whether to operate less efficiently to keep propellant production high, decrease the production rate to conserve energy, or request that parts be cleaned or replaced. Throughout the mission, crewmember or ground personnel may adjust how the autonomous system makes these decisions.

DS3

Deep Space Three (DS3) is a set of three spacecraft to be launched in a single vehicle in 2001. The mission goal is to test a large optical interferometer formed by the three spacecraft. The purpose of developing large spaceborne interferometer is to discover earth-sized planets around other stars. In space, the three spacecraft will form an equilateral triangle with each side from 100m to 1 km depending on the accuracy desired. The spacecraft must maintain their positions relative to each other with accuracy of +- 1 cm. The spacecraft are expected to target 50 stars during its 6-month mission.

This mission is interesting from an autonomous system viewpoint because three spacecraft must be precisely coordinated. In addition, the propellant on each spacecraft limits the life of the mission. The quantity on each spacecraft may vary considerably as the mission progresses. Once the propellant of one spacecraft is exhausted, they will no longer be able to form an equilateral triangle. To address this problem, the resource manager of the autonomous system will attempt to compensate. For example, if one spacecraft is low on propellant, the autonomous system may instruct the other two to use more propellant to reduce the propellant consumed by the one that is low. Moreover, scientists will want to aim these spacecraft like they would a large telescope. The plans must be made so that the most important images can be taken while minimizing the propellant consumed. This combination of autonomous operation and human interaction makes it an excellent candidate for an autonomous system that supports adjustable autonomy.

DS4

Deep Space Four (DS4) is a spacecraft with a 200kg lander that will land on the comet Tempel 1. The lander will take images of the comet, collect and analyze samples up to one meter below the surface. The lander will return to Earth with up to 100 cubic centimeters of comet material.

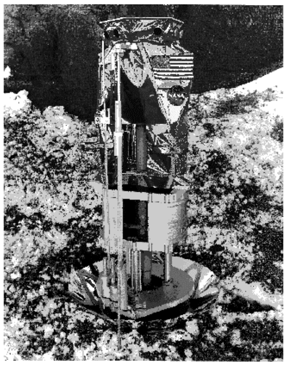


Figure 7. Drawing of Deep Space 4 Lander Prototype

Although this science goal is laudable, the primary mission goal is to test advanced technologies, including autonomous control systems, necessary for landing spacecraft on small bodies in space, e.g., asteroids. NASA scheduled the DS4 launch for 2003, the comet rendezvous in 2005, and return to Earth in 2010.

This mission presents an interesting challenge for autonomy. Scientists and spacecraft designers have only a vague idea of what to expect when the lander attempts to land on a comet. During the landing sequence, the lander must operate autonomously due to the delay of the radio signal to Earth and back. The lander must appropriately react to whatever state it finds itself in, including hardware failures due to flying into the comet's tail and impacting the comet. Once on the comet, scientists would like to give specific commands to the lander based on data transmitted from the lander. However, the lander must complete its mission even if it loses its communication link to Earth and return with a comet sample.

Reusable Launch Vehicles (Shuttle, X33, VentureStar)

The projects mentioned above have focused on autonomous systems in space and on Mars. However, autonomous systems will play an essential role launching people and material from Earth to space and will reducing the cost and risk of Mars missions.

Figure 8. Drawing of X-33

Autonomous systems are being considered to manage ground operations for reusable launch vehicles, in particular, the Space shuttle, X-33, and VentureStar. The autonomous system will collect data from the spacecraft while it is in flight and determine what needs to be maintained. Unlike, the previous autonomous systems, this autonomous system will rely on many people to execute its maintenance plan. When people or the autonomous system discovers additional problems or if maintenance and repairs are not made according to schedule, the autonomous system creates a recovery plan to stay on schedule and in budget as guided by the constraints human managers place on it.

Summary

NASA is enagaged in several projects that "push the envelope" on the design of autonomous systems that support adjustable autonomy. Current NASA autonomous systems include the Remote Agent and 3T. The Remote Agent was designed for controlling unmanned spacecraft and rover for extended periods of time while managing consumable resources throughout a mission and using model-based diagnosis and recovery to handle failures. 3T was designed to interact with humans. This includes control by humans at various levels and controlling systems in environments with humans, e.g., life-support system and a robot working with a human. Future space missions will require autonomous systems with features from both of these autonomous systems. Crewmembers will operate and maintain several systems, from life support, to fuel production, to rovers. Without autonomy, they would be spending most of their time just trying to stay alive. However, even with completely autonomous systems, crewmembers would be frustrated by how to repair them, how to get them to do exactly what they want, and even to understand why the systems are behaving as they do or predicting how the systems will behave under certain conditions. Adjustable autonomy is essential to address these problems and vital for autonomous systems that interact with people.

Acknowledgements

The authors acknowledge the efforts of Chuck Fry, Bob Kanefsky, Ron Keesing, Jim Kurien, Bill Millar, Nicola Muscettola, Pandu Nayak, Chris Plaunt, Kanna Rajan, Brian Williams of NASA Ames; Douglas Bernard, Ed Gamble, Erann Gat, Nicolas Rouquette, and Ben Smith of JPL for their efforts in developing the Remote

Agent. The authors also acknowledge Cliff Farmer, Dan Poirot, Dan Ryan, Carroll Thronesbery, Mary Beth Edeen , and Karen Meyers for their support of the work using 3T at JSC. Images of the DS1, DS4, X33 spacecraft and Marsokhod rover were provided by NASA.

References

[Allen, 1994] Allen, J. Mixed initiative planning: Position paper. *ARPA/Rome Labs Planning Initiative Workshop,* Feb. 1994.

[Bonasso *et al.*, 97a] Bonasso, R. P., D. Kortenkamp, and T. Whitney. Using a robot control architecture to automate space station shuttle operations. *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, Cambridge, Mass., AAAI Press, 1997.

[Bonasso *et al* 97b] Bonasso, R. P., R. J. Firby, E. Gat, D. Kortenkamp, D. Miller, M. Slack. Experiences with an Architecture for Intelligent, Reactive Agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 9(2), 1997.

[Bonasso *et al.*, 98] Bonasso, R. P., R. Kerr, K. Jenks, and G. Johnson. Using the 3T Architecture for Tracking Shuttle RMS Procedures, *IEEE International Joint Symposia on Intelligence and Systems (SIS)*, Washington D.C., May 1998.

[Bernard *et al.*, 98] Bernard, D. E., G. A. Dorais, C. Fry, E. B. Gamble Jr., B. Kanefsky, J. Kurien, W. Millar, N. Muscettola, P. P. Nayak, B. Pell, K. Rajan, N. Rouquette, B. Smith, and B. C. Williams. Design of the Remote Agent experiment for spacecraft autonomy. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, 1998.

[Cohen, 93] Cohen, M. Mars surface habitation study. Presentation at the Mars Exploration Study Workshop II, NASA Conference Publication 3243. Conference held at NASA Ames Research Center, May 24-25, 1993.

[Elsaesser and MacMillan, 1991] Elsaesser, C., and E. MacMillan. Representations and Algorithms for Multi Agent Adversarial Planning, MITRE Technical Report MTR-199191W000207, 1991.

[Firby, 1989] Firby, R. J. *Adaptive Execution in Complex Dynamic Worlds*. Ph.D. Thesis, Yale University, 1989.

[Gat, 1998] Gat, E. Three-Layer Architectures, *Artificial Intelligence and Mobile Robots,* D. Kortenkamp, R. P. Bonasso, and R. Murphy, eds., AAAI/MIT Press, Cambridge, MA, 1998.

[Hoffman and Kaplan, 97] Hoffman, S. J., and D. I. Kaplan, Eds. *Human Exploration of Mars: The Reference Mission of the NASA Mars Exploration Study Team.* NASA Special Publication 6107, Johnson Space Center: Houston, TX, July 1997.

[Kortenkamp *et al.*, 97] Kortenkamp, D., P. Bonasso, D. Ryan, and D. Schreckenghost. Traded control with autonomous robots as mixed initiative interaction. AAAI-97 Spring Symposium. *Workshop on Mixed Initiative Interaction,* Mar 1997.

[Lewis *et al.*, 1998] Lewis, J. F., N. J. C. Packham, V. L. Kloeris, and L. N. Supra. The Lunar-Mars Life Support Test Project Phase III 90-day Test: The Crew Perspective. *28th International Conference on Environmental Systems*, July 1998.

[Muscettola *et al.*, 98] Muscettola, N., P. P. Nayak, B. Pell, and B. C. Williams. Remote Agent: To boldly go where no AI system has gone before. *Artificial Intelligence*, 103(1/2), August 1998. To appear.

[Pell *et al.*, 96] Pell, B., D. Bernard, S. A. Chien, E. Gat, N. Muscettola, P. P. Nayak, M. D. Wagner, and B. C. Williams. A remote agent prototype for spacecraft autonomy. In *Proceedings of the SPIE Conference on Optical Science, Engineering, and Instrumentation*, 1996.

[Pell *et al.*, 98] Pell, B., S. Sawyer, D. E. Bernard, N. Muscettola, and B. Smith. Mission operations with an autonomous agent. In *Proceedings of the IEEE Aerospace Conference*, Snowmass, CO, 1998.

[Pell *et al.*, 99] Pell, B., G. A. Dorais, C. Plaunt, R. Washington. The Remote Agent Executive: Capabilities to Support Integrated Robotic Agents. Submitted to *Autonomous Robotics Journal,* for publication in 1999.

[Plaunt and Rajan, 98] Plaunt, C., and K. Rajan. IpexT: integrated planning and execution for military satellite tele-communications. In *Working Notes of the AIPS Workshop*, 1998.

[Schreckenghost and Thronesbery, 1998] Schreckenghost, D. and C. Thronesbery. Integrated Display for Supervisory Control of Space Operations. *Human Factors and Ergonomics Society*. 42nd Annual Meeting, Chicago, IL, October 1998.

[Schreckenghost *et al.*, 98] Schreckenghost, D., D. Ryan, C. Thronesbery, P. Bonasso, and D. Poirot. Intelligent control of life support systems for space habitats. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Conference on Innovative Applications of Artificial Intelligence*. Madison, WI, AAAI Press, pp. 1140-1145, July 1998.

[Schreckenghost *et al.*, 98a] Schreckenghost, D., M. Edeen, P. Bonasso, and J. Erickson. Intelligent control of product gas transfer for air revitalization. In *Proceedings of 28th International Conference on Environmental Systems.* Danvers, MA, July 1998.

[Williams and Nayak, 96] Williams, B. C., and P. P. Nayak. A model-based approach to reactive self-configuring systems. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence.* Cambridge, Mass., AAAI Press, pp. 971-978, 1996.